

174/286
(A1081)

CONFIGURABLE CROSSBAR SWITCH

Background of the Invention

[0001] This invention relates to programmable logic resources. More particularly, this invention relates 5 to providing more flexible connectivity between signaling input-output (I/O) and an intellectual property block in a programmable logic resource.

[0002] Programmable logic resource technology is well known for its ability to allow a common hardware 10 design (embodied in an integrated circuit) to be programmed to meet the needs of many different applications. Known examples of programmable logic resource technology include programmable logic devices (PLDs), complex programmable logic devices (CPLDs), 15 erasable programmable logic devices (EPLDs), electrically erasable programmable logic devices (EEPLDs), and field programmable gate arrays (FPGAs).

[0003] A programmable logic resource is typically 20 embedded on a silicon chip that sits in a package containing pins. The programmable logic resource has signaling I/O ports along the periphery of the programmable logic resource that allows data to be

received from and transmitted to the pins on the package. The package is typically mounted on a circuit board, which allows for the programmable logic resource to communicate with circuitry external to the package.

5 The circuit board contains wiring that connects the pins to other components on the circuit board.

[0004] To facilitate the use of programmable logic resources in certain applications, intellectual property (IP) blocks are coupled to programmable logic resource core circuitry. Data from circuitry external to the package is typically sent to a programmable logic resource through a particular pin and to a corresponding I/O port where the data is decoded and sent to a corresponding data port in the IP block for processing. Similarly, data from the IP block is typically sent to circuitry external to the package through a data port to a corresponding I/O port where the data is encoded and sent to a corresponding pin for output.

10 15 20 25 30 [0005] During the design stage of a programmable logic resource, a user may not know the size of the programmable logic resource. Often during the same time, a board vendor will design a circuit board having fixed pin connections that support a particular package in which a programmable logic resource is to be located. Because the circuit board is designed with fixed pin connections, the resulting programmable logic resource will be placed in the particular package for which the circuit board is designed to support. However, with different possible sizes of programmable logic resources, the locations of the I/O ports and corresponding data ports at the IP block of the programmable logic resource may change relative to the

pin locations for a given package, thereby causing connectivity problems. The data ports in the IP block cannot be easily reconfigured so that the data ports and corresponding I/O ports are matched up with the 5 correct pins without incurring additional costs. In addition, the circuit board cannot be easily rewired so that the pins are matched up with the correct I/O ports and corresponding data ports without incurring additional costs. As a result, extra wiring is 10 typically used to route data between a particular pin to a corresponding I/O port. However, this extra wiring can be limited by the available space for the extra wiring, can cause additional delay times, and can cause further connectivity problems.

15 [0006] In view of the foregoing, it would be desirable to provide more flexible connectivity (i.e., vertical migration) between signaling I/O and an intellectual property block in a programmable logic resource.

20 Summary of the Invention

[0007] In accordance with the invention more flexible connectivity (i.e., vertical migration) between signaling input-output (I/O) and an intellectual property (IP) block in a programmable 25 logic resource is provided.

[0008] Vertical migration is achieved by providing a configurable crossbar switch between the signaling I/O and the IP block in a programmable logic resource. A programmable logic resource receives input data via an 30 I/O port. This data is decoded in an I/O buffer and sent as input to a crossbar switch that can be configured to send the data to any one of the data

ports in the IP block. Similarly, data from the IP block can be sent via a data port to a crossbar switch that can be configured to send the data to an I/O buffer that encodes the data for output to any one of 5 the I/O ports. The crossbar switch can be configured based on different factors including, for example, the design of the programmable logic resource, the package in which the programmable logic resource is to sit, and the routing from the package pins to the other 10 components in a circuit board. Because of the flexibility in connecting any of the I/O ports in the I/O buffer to any of the data ports in the IP block, each package pin can be routed to the best possible I/O port rather than to the I/O port that directly 15 corresponds to a particular data port in the IP block.

[0009] The use of a crossbar switch advantageously provides greater flexibility in the design of a programmable logic resource. The invention also advantageously reduces connectivity problems between 20 the signaling I/O and an IP block. The invention further reduces the amount of routing needed while not requiring the IP block to be reconfigured or the circuit board to be rewired in order to correctly connect the signaling I/O to the IP block.

25 Brief Description of the Drawings

[0010] The above and other objects and advantages of the invention will be apparent upon consideration of the following detailed description, taken in conjunction with the accompanying drawings, in which 30 like reference characters refer to like parts throughout, and in which:

[0011] FIGS. 1-2 are simplified diagrams of a circuit board;

[0012] FIG. 3 is a simplified diagram of a circuit board in accordance with the invention;

5 [0013] FIG. 4 is a simplified block diagram of a programmable logic resource core in accordance with the invention;

[0014] FIGS. 5-7 are simplified block diagrams of a crossbar switch in accordance with the invention; and

10 [0015] FIG. 8 is a simplified schematic block diagram of a system employing a programmable logic resource, multi-chip module, or other suitable device in accordance with the invention.

Detailed Description

15 [0016] In accordance with the invention, more flexible connectivity (i.e., vertical migration) between signaling input-output (I/O) and an intellectual property (IP) block in a programmable logic resource is provided. A crossbar switch is provided between the signaling I/O and the IP block in a programmable logic resource. A programmable logic resource receives input data via an I/O port. This data is decoded in an I/O buffer and sent as input to a crossbar switch that can be configured to send the data to any one of the data ports in the IP block.

20 Similarly, data from the IP block can be sent via a data port to a crossbar switch that can be configured to send the data to an I/O buffer that encodes the data for output to any one of the I/O ports. The crossbar switch can be configured based on different factors

25 including, for example, the design of the programmable logic resource, the package in which the programmable

logic resource is to sit, and the routing from the package pins to the other components in a circuit board. Because of the flexibility in connecting any of the I/O ports in the I/O buffer to any of the data ports in the IP block, each package pin can be routed to the best possible I/O port rather than to the I/O port that directly corresponds to a particular data port in the IP block.

5 [0017] A board vendor will design a circuit board 10 having different components that are routed together to allow data to flow between the different components. The different components can include logic gates, 15 circuits designed to perform particular functions, programmable logic resources, memories, processors, input/output circuitry, peripheral devices, and any 20 other suitable components. Some of these different components are typically placed in a package that has external pins to allow data to flow into and output of the component. Each package can be mounted onto the circuit board at designated locations, with the pins 25 making contact with the circuit board at fixed pin locations. Wiring is used to route data between the fixed pin locations and other components on the circuit board.

25 [0018] FIG. 1 is a partial diagram of a circuit board 100. A package 110 containing a programmable logic resource 120 is mounted on circuit board 100. Package 110 has external pins 112 that make contact 30 with circuit board 100 at fixed pin locations. Circuit board 100 includes wiring 130 used to route data between each pin 112 to a designated location throughout circuit board 100. Data can be sent between programmable logic resource 120 and circuit board 100

via pins 112, wiring 114, and I/O ports 122 located along the periphery of programmable logic resource 120. The data sent along wiring 130 and wiring 114 can be sent via any suitable signals including single-ended I/O signals and differential I/O signals (e.g., low voltage differential signaling (LVDS) I/O).

5 [0019] FIG. 2 is a partial diagram of circuit board 100' when a larger programmable logic resource 220 is provided. Programmable logic resource 220 is placed in the same package 110 for which circuit board 100' is designed to support. Pins 112 make contact with circuit board 100' at the same fixed pin locations. Circuit board 100' also includes the same wiring 130 used to route data between 10 each pin 112 to a designated location throughout circuit board 100'. Because a larger programmable logic resource 220 is provided, the locations of the I/O ports 222 in programmable logic resource 220 are no longer aligned with the locations of corresponding 15 pins 112. In an existing approach, extra wiring 214 is needed to connect I/O port 222 to a corresponding pin 112.

20 [0020] To provide more flexible connectivity, a crossbar can be implemented in the programmable logic resource to allow any suitable package pin to be connected to any suitable I/O port on a programmable logic resource without requiring each package pin to be connected to a corresponding I/O port on the programmable logic resource. FIG. 3 is a partial 25 diagram of circuit board 100" when a larger programmable logic resource 320 is provided in accordance with the invention. Instead of using extra 30 wiring 214 to connect I/O port 222 to a corresponding

pin 112 as shown in FIG. 2, wiring 314 can be used to connect a given pin 112 to the closest available I/O port 322.

[0021] Circuitry such as a crossbar switch can be provided in programmable logic resource 320 to allow data to flow between an I/O port 322 and a corresponding data port in an IP block as shown in FIG. 4. A programmable logic resource (e.g., resource 320) can include a programmable logic resource core 400, I/O buffers 410, a crossbar switch 420, and an IP block 430. Programmable logic resource core 400 can be typical programmable logic circuitry of any of several known types and constructions. IP block 430 can be a soft IP block in which circuitry is programmed in as programmable logic. Circuitry is described in terms of gates and interconnections for the gates using a netlist or a hardware description language (e.g., Very High Speed Integrated Circuit Hardware Description Language (VHDL) or Verilog), which can be synthesized with the surrounding logic. In another embodiment, IP block 430 can be a hard IP block in which circuitry is custom-designed in hardware. In yet another embodiment, IP block 430 can be a firm IP block in which circuitry is partly designed in hardware but is configurable for various applications.

[0022] Although the invention is described herein primarily in the context of a crossbar switch on a programmable logic resource for clarity, the crossbar switch can be provided on any suitable device such as, for example, an application-specific standard product (ASSP), an application-specific integrated circuit (ASIC), a full-custom chip, or a dedicated chip.

[0023] Data can be transmitted along paths 402 (e.g., paths 314 and path 130) between programmable logic resource core 400 and other circuitry external programmable logic resource core 400. Programmable logic resource core 400 can receive data from paths 402 via I/O ports located along the periphery of the programmable logic resource core 400 at I/O buffers 410. I/O buffers 410 decode the input data and output the decoded data to a crossbar switch 420 via paths 412. Crossbar switch 420 can be configured to send the decoded data to any one of the data ports in IP block 430 via paths 422 for processing.

Programmable logic resource core 400 can also transmit data onto paths 402. IP block 420 can send data through data ports to crossbar switch 420. Crossbar switch 420 can be configured to send the data to I/O buffers 410 via any one of paths 412. I/O buffers 410 encodes the data and outputs the encoded data onto paths 412 via the I/O ports. IP block 430 can send data to and receive data from programmable logic resource core 400 via paths 432.

[0024] In one embodiment, the crossbar switch can be a bi-directional switch. In another embodiment, the crossbar switch can be a uni-directional switch. To provide the flow of data between IP block 430 and I/O buffers 410, two uni-directional switches can be implemented. The configuration of the crossbar switch for the transmission of data from I/O buffers 410 to IP block 430 can be the same as or different from the configuration of the crossbar switch for the transmission of data from IP block 430 to I/O buffers 410.

[0025] FIGS. 5-7 are simplified diagrams of a crossbar switch 500 in accordance with the invention. Crossbar switch 500 (e.g., crossbar switch 420) includes crossbar circuitry 510 and control circuitry 520. In one embodiment, crossbar circuitry 510 receives input data from I/O buffers (e.g., I/O buffers 410) via paths 512 (e.g., paths 412). Crossbar circuitry 510 can be configured to route each input data to any one of paths 514 (e.g., paths 422) for output to an IP block (e.g., IP block 430). In another embodiment, crossbar circuitry 510 receives input data from an IP block (e.g., IP block 430) via paths 514 (e.g., paths 422). Crossbar circuitry 510 can be configured to route each input data to any one of paths 512 (e.g., paths 412) for output to I/O buffers (e.g., I/O buffers 410).

[0026] Control circuitry 520 can receive as input a control signal 522 from programmable logic resource core 400, IP block 430, or any component on or external to circuit board 300 (e.g., another programmable logic resource, processor, memory, user input) via I/O buffers 410. Control circuitry 520 can, based on control signal 522, send data for configuration to crossbar circuitry 510 via path 524. Alternatively, control signal 522 can be sent directly to crossbar circuitry 510 for configuration. Control signals 522 can indicate the proper connections between paths 512 and paths 514 using any suitable approach.

[0027] FIG. 6 illustrates a configuration of a crossbar switch 600 when a programmable logic resource has been designed such that the I/O ports are properly aligned to the corresponding package pins (e.g., FIG. 1). Connections can be provided between the

package pins and I/O ports such that data can be directly sent to and from corresponding data ports in an IP block. Crossbar circuitry 610 can be configured such that each path 512 is coupled to a directly 5 corresponding path 514.

[0028] FIG. 7 illustrates a configuration of a crossbar switch 700 when a programmable logic resource has been designed such that the I/O ports are not properly aligned to the corresponding package pins 10 (e.g., FIG. 3). Crossbar circuitry 710 can be configured to switch the connections between paths 512 and paths 514 such that each path 512 is coupled to a corresponding path 514.

[0029] A crossbar switch can be configured at any 15 suitable time such as, for example, upon final design of a programmable logic resource, during initial configuration of the programmable logic resource, or during reconfiguration of the programmable logic resource. In another embodiment, to support changes in 20 the routing of signals, the crossbar switch can be designed to be dynamically adjustable such that the crossbar switch can support different configurations while data is processing in the programmable logic resource or at another suitable time. Providing a 25 configurable crossbar switch allows flexibility in the design of a programmable logic resource and also reduces connectivity problems.

[0030] FIG. 8 illustrates a programmable logic resource 802, multi-chip module 804, or other device (e.g., ASSP, ASIC, full-custom chip, dedicated chip). which includes embodiments of this invention in a data processing system 800. Data processing system 800 can include one or more of the following components: a

processor 806, memory 808, I/O circuitry 810, and peripheral devices 812. These components are coupled together by a system bus or other interconnections 820 and are populated on a circuit board 830 which is contained in an end-user system 840.

[0031] System 800 can be used in a wide variety of applications, such as computer networking, data networking, instrumentation, video processing, digital signal processing, or any other application where the 5 advantage of using programmable or reprogrammable logic is desirable. Programmable logic resource/module 802/804 can be used to perform a variety of different logic functions. For example, programmable logic resource/module 802/804 can be configured as a 10 processor or controller that works in cooperation with processor 806. Programmable logic resource/module 802/804 may also be used as an arbiter for arbitrating access to a shared resource in system 800. In yet another example, programmable logic resource/module 15 802/804 can be configured as an interface between processor 806 and one of the other components in system 800. It should be noted that system 800 is only exemplary, and that the true scope and spirit of the invention should be indicated by the following claims.

[0032] Various technologies can be used to implement 20 programmable logic resources 802 or multi-chip modules 804 having the features of this invention, as well as the various components of those devices (e.g., programmable logic connectors ("PLCs") and programmable 25 function control elements ("FCEs") that control the PLCs). For example, each PLC can be a relatively simple programmable connector such as a switch or a plurality of switches for connecting any one of several

inputs to an output. Alternatively, each PLC can be a somewhat more complex element that is capable of performing logic (e.g., by logically combining several of its inputs) as well as making a connection. In the 5 latter case, for example, each PLC can be a product term logic, implementing functions such as AND, NAND, OR, or NOR. Examples of components suitable for implementing PLCs include EPROMs, EEPROMs, pass transistors, transmission gates, antifuses, laser 10 fuses, metal optional links, etc. PLCs and other circuit components may be controlled by various, programmable, function control elements ("FCEs"). For example, FCEs can be SRAMS, DRAMS, magnetic RAMS, ferro-electric RAMS, first-in first-out ("FIFO") 15 memories, EPROMS, EEPROMs, function control registers, ferro-electric memories, fuses, antifuses, or the like. From the various examples mentioned above it will be seen that this invention is applicable to both one-time-only programmable and reprogrammable resources. 20 [0033] Thus it is seen that providing a configurable crossbar switch advantageously allows flexibility in the design of a programmable logic resource and reduces connectivity problems. One skilled in the art will appreciate that the invention can be practiced by other 25 than the prescribed embodiments, which are presented for purposes of illustration and not of limitation, and the invention is limited only by the claims which follow.